

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

---

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

2013

Michal Walder



## Zadání bakalářské práce

Student: **Michal Walder**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**  
**Individual Professional Practice in the Company**

### Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: CATHEDRAL Software, s.r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
  - c) Zvolený postup řešení zadaných úkolů
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

### Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Doc. Mgr. Jiří Dvorský, Ph.D.**

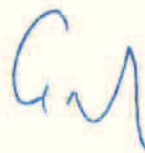
Konzultant bakalářské práce: Ing. Josef Šustr

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty



Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. května 2013

.....*Ka. Dep*.....



Rád by som sa touto formou poďakoval Ing. Josefovi Šustrovi za umožnenie vykonávania odbornej praxe vo firme Cathedral Software, s.r.o., za odbornú pomoc a konzultácie počas celej doby odbornej praxe.

Vďaka patrí aj Ing. Petrovi Míčkovi za jeho pomoc a spoluprácu pri riešení rôznych problémov.

Ďakujem taktiež doc. Mgr. Jiřímu Dvorskému, Ph.D. za odbornú pomoc pri spracovaní mojej bakalárskej práce.





## Abstrakt

Táto bakalárská práca popisuje priebeh mojej individuálnej odbornej praxi v softvérovej firme Cathedral Software, s.r.o.. V úvodnej časti práce uvádzam profil firmy, jej zameranie a produkty. Dozviete sa podrobné informácie o vyvíjanom produkte a platforme, na ktorej je vyvíjaný. Podrobne popisujem technológie, ktoré som používal pri práci na projekte. Ďalej rozoberám vybrané úlohy vykonávané počas odbornej praxe a znalosti, ktoré som pri nich zúročil, ale aj tie, ktoré som pri nich nadobudol. Prevažne sa zaoberám mojou hlavnou úlohou - a to vývojom mobilnej aplikácie pre informačný systém ArisCat. Na záver zhodnocujem teoretické a praktické prínosy a nedostatky odhalené počas odbornej praxe.

**Kľúčová slova:** Odborná praxe, Cathedral Software, Chytrý telefón, Mobilná aplikácia, ArisCat, HTML5 Builder, Eclipse, Java, Android, XML

## Abstract

This bachelor thesis describes my experiences during my individual professional practice at the software company of Cathedral Software Ltd.. In the introductory part of the thesis I have introduced the companys profile, its focus and products. I have provided a detailed description of the product in development and the platform on which it has been developed. I have also given a detailed description of the technologies that I have used while was working on the project. Subsequently, I have analyzed the selected assignments I was working during my practice and knowledges that I used, but also those, that I acquired. Mostly I have focused on my main assignment and its development of a mobile application for the information system ArisCat. Finally, I have review the theoretical and practical benefits of my practice as well as pointed out the lack of some technical knowledge on my behalf that I have discovered during the practice.

**Keywords:** Professional practice, Cathedral Software, Smart phone, Mobile application, ArisCat, HTML5 Builder, Eclipse, Java, Android, XML



## Seznam použitých zkratek a symbolů

ArisCAT	– Informačný systém spoločnosti Cathedral Software s.r.o.
Framework	– Software/Hardware framework (vývojová platforma)
GUI	– Graphical User Interface (grafické užívateľské rozhranie napr. aplikácie)
HTTP	– HyperText Transfer Protocol (protokol pre prenos html dokumentov medzi servermi a klientmi služby WWW)
IDE	– Integrated Development Environment (vývojové prostredie, obsahujúce editor zdrojového kódu, kompilátor a väčšinou aj debugger)
IS	– Information system (informačný systém)
Java	– Programming language (programovací jazyk)
open-source (software)	– Počítačový softvér s otvoreným zdrojovým kódom, čo znamená, že používatelia musia mať právo voľne ho používať, modifikovať a šíriť.
plugin	– Počítačový program, ktorý rozširuje funkcie iného programu
PHP	– Hypertext preprocessor (skriptovací programovací jazyk)
SDK	– Software Developer Kit (súbor vývojárskych nástrojov)
user-friendly	– Používateľsky prívetivé (napr. prostredie aplikácie a pod.)
XML	– Extensible Markup Language (obecný značkovací jazyk)



## Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Zasadenie odbornej praxe</b>	<b>9</b>
2.1	Profil firmy Cathedral Software s.r.o. . . . .	9
2.2	Popis pracovného zaradenia . . . . .	10
<b>3</b>	<b>Zadané úlohy v priebehu praxe</b>	<b>11</b>
3.1	Zoznámenie s IS ArisCat . . . . .	11
3.2	Štúdie : Možnosti vývoja mobilných aplikácií . . . . .	11
3.3	Tvorba GUI pre mobilnú aplikáciu v HTML5 Builderi . . . . .	11
3.4	Pridávanie funkcionality do vytvoreného GUI . . . . .	11
3.5	Testovanie vytvorenej aplikácie na reálnych zariadeniach . . . . .	11
3.6	Štúdie: Vývoj pre Android vs. iOS . . . . .	11
3.7	Vývoj aktivity pre prihlasovanie . . . . .	12
3.8	Vývoj testovacej aplikácie pre testovanie ovládateľnosti aplikácie . . . . .	12
3.9	Dynamická XML maska . . . . .	12
3.10	Vývoj aktivity pre dynamické generovanie obsahu . . . . .	12
3.11	Graf časovej náročnosti jednotlivých úloh . . . . .	13
<b>4</b>	<b>Používané technológie</b>	<b>15</b>
4.1	HTML5 Builder . . . . .	15
4.2	PhoneGap . . . . .	15
4.3	Eclipse IDE . . . . .	15
4.4	Java . . . . .	16
4.5	XML . . . . .	16
4.6	Android SDK . . . . .	17
<b>5</b>	<b>Zvolený postup riešenia zadaných úloh</b>	<b>19</b>
5.1	Zoznámenie s IS ArisCAT . . . . .	19
5.2	Štúdie : Možnosti vývoja mobilných aplikácií . . . . .	20
5.3	Tvorba GUI pre mobilnú aplikáciu v HTML5 Builderi . . . . .	20
5.4	Pridávanie funkcionality do vytvoreného GUI . . . . .	20
5.5	Testovanie vytvorenej aplikácie na reálnych zariadeniach . . . . .	21
5.6	Štúdie: Vývoj pre Android vs. iOS . . . . .	22
5.7	Vývoj aktivity pre prihlasovanie . . . . .	22
5.8	Vývoj testovacej aplikácie pre testovanie ovládateľnosti aplikácie . . . . .	23
5.9	Dynamická XML maska . . . . .	24
5.10	Vývoj aktivity pre dynamické generovanie obsahu . . . . .	25
<b>6</b>	<b>Potrebné znalosti získané počas štúdia potrebné pre výkon odbornej praxe</b>	<b>27</b>
<b>7</b>	<b>Odhalené nedostatky počas absolvovania praxe</b>	<b>29</b>

<b>8</b>	<b>Hodnotenie bakalárskej praxe</b>	<b>31</b>
<b>9</b>	<b>Literatúra</b>	<b>33</b>
<b>10</b>	<b>Prílohy</b>	<b>35</b>

## Zoznam obrázkov

1	Logo firmy Cathedral Software s.r.o. . . . .	9
2	Graf časovej náročnosti jednotlivých úloh . . . . .	13
3	Zadávanie objednávky vo webovom IS ArisCAT . . . . .	19
4	Prihlasovacia aktivita . . . . .	22
5	Hlavné menu aplikácie . . . . .	23
6	Výsledný vzhľad dynamicky vygenerovanej aktivity . . . . .	26





## Seznam výpisů zdrojového kódu

1	Ukážka funkcí využívaných pre zobrazovanie/skrytie elementov GUI . .	21
2	Ukážka funkcí využívaných pre povoľovanie/zakazovanie elementov GUI	21
3	Ukážka metódy generujúcej XML dokument . . . . .	25



## 1 Úvod

Bakalárska práca popisuje absolvovanie individuálnej odbornej praxe v softvérovej firme z pohľadu študenta informatiky.

Odbornú prax som vykonával vo firme Cathedral Software s.r.o. čo je firma poskytujúca služby v oblasti IT. Túto prax som vykonával počas 50 pracovných dní v zimnom a letnom semestri akademického roku 2012/2013. Za túto dobu som získal množstvo cenných pracovných skúseností a poznatkov, ktoré som zúročil pri vlastnom štúdiu. Zároveň som pri riešení rôznych úloh využil veľké množstvo teoretických a praktických poznatkov získaných počas štúdia.

V práci je popísané odborné zameranie firmy Cathedral Software s.r.o. a moje zaradenie v programátorskom tíme, ktoré som zastával. Ďalšie časti práce sa zaoberajú popisom projektu, konkrétnymi pracovnými úlohami na ktorých som pracoval a postupmi, ktoré som pri riešení úloh zvolil. Jedna časť práce sa venuje popisu technológií a softwaru ktorý som pri práci používal.

Na záver ponúknem hodnotenie svojich výsledkov na základe nadobudnutých odborných znalostí. Popíšem teoretické a praktické znalosti ktoré som počas odbornej praxe získal a nedostatky, ktoré sa objavili pri práci na riešených úlohách.



## 2 Zasadenie odbornej praxe

### 2.1 Profil firmy Cathedral Software s.r.o.

Spoločnosť Cathedral Software vznikla v polovici roku 1996 ako združenie fyzických osôb Cathedral Software and Hardware a ich sídlo bolo vždy v Prostějově. Firma disponuje rýdzo českým kapitálom a ich orientácia bola od vzniku zameraná na dlhodobú spoluprácu spojenú s tvorbou kvalitného softwaru na zákazku, a poskytovanie ostatných služieb podľa požiadaviek zákazníkov. V súvislosti s požiadavkami zákazníkov firma rozšírila portfólio aj o dodávku počítačových komponentov a zostáv. V rokoch 1997-1998 začala s dodávkami, inštaláciami a správou informačných systémov.



Obr. 1: Logo firmy Cathedral Software s.r.o.

Pracovníci firmy Cathedral Software sa najintenzívnejšie zameriavali vždy na vývoj informačných systémov tvorených podľa prání a potrieb zákazníkov. Ako jeden z prvých bol informačný systém Žalúzie, ktorý bol určený pre výrobcov žalúzií. Ďalej potom Hokej 2000, ktorý bol vytvorený s podporou Svazu ledního hokeje ČR.

Ďalej firma značne spolupracuje so spoločnosťou LCS International a.s., ktorá patrí medzi najväčších producentov podnikových aplikácií na českom trhu. Cieľom tejto spolupráce je vývoj a následná distribúcia integrovaného modulu pre komunikáciu s informačným systémom Helios, ktorý bude zaisťovať spracovanie obchodov s ťažko konfigurovateľnými výrobkami, ako sú napríklad žalúzie, rolety, vráta a plastové okná.

Zlomový informačný systém bol bezpochyby IS ArisCAT, ktorým sa firma zaoberá už od roku 2001. Tento systém je firmou vyvíjaný dodnes a predovšetkým je optimalizovaný na spracovanie už vyššie spomínaných, ťažko konfigurovateľných výrobkov. S týmto systémom som pracoval a vyvíjal som jeho ekvivalent pre mobilné zariadenia. Detailnejším informáciám a funkčnosti tohto systému bude venovaná jedna z ďalších kapitol. [1]

## 2.2 Popis pracovného zaradenia

Pri snahe o získanie odbornej praxe, bolo nutné zúčastniť sa osobného pohovoru s vedúcim oddelenia vývoja softwaru. Na pohovore boli preverené moje znalosti v technológiách a programovacích jazykoch na základe môjho zaslaného životopisu. Po úspešnom prijatí do firmy nasledovalo druhé stretnutie, kde som bol informovaný hlavne o technickom zameraní spoločnosti, o používaných technológiách a metódach. Mal som možnosť vidieť pracovné prostredie, spôsob komunikácie a technické vybavenie spoločnosti.

Na základe mojich znalostí mi bola pridelená práca vývojára mobilnej aplikácie. Najskôr bol vývoj orientovaný pre všetky mobilné platformy, ale na základe zistení v priebehu práce sa vývoj preorientoval na mobilnú platformu Google Android. Práca mala charakter niekoľkých úloh. Na niektorých úlohách som pracoval samostatne, no väčšinou úlohy vyžadovali tímovú prácu a aktívnu komunikáciu medzi členmi vývojárskeho tímu. Úlohy zahŕňali problematiku od štúdií, konzultácií a zisťovania informácií pre zaprázdnený vývojársky tím až po samotný vývoj aplikácie.

Na začiatku som bol oboznámený so samotnou firmou, jej produktami a službami. Neskôr mi bol predstavený IS ArisCat. Práve tento systém som vyvíjal pre mobilné zariadenia, takže nasledovalo úplne zoznámenie so systémom a vysvetlenie funkcionality jednotlivých častí systému. Vedúci vývojárskeho tímu mi pri tejto príležitosti popísal požiadavky firmy na mobilnú aplikáciu, na ktorej som počas praxe pracoval.

Pri práci na zadanych úlohách som bol podporovaný zamestnancami, ktorí boli veľmi ochotní a komunikácia medzi mnou a vývojárskym tímom bola na výbornej úrovni.

### 3 Zadané úlohy v priebehu praxe

Počas priebehu praxe mi bolo zadaných celkovo 10 čiastkových úloh. Práca začínala zoznámením s IS ArisCat, pokračovala štúdiom možností vývoja mobilných aplikácií až k samotnému vývoju. Pre vývoj bol zvolený HTML5 Builder od firmy Embarcadero, no neskôr pre problémy s testovaním na reálnych zariadeniach bol zvolený vývoj v Eclipse IDE so štandardným Android SDK.

#### 3.1 Zoznámenie s IS ArisCat

Prvou úlohou bolo zoznámenie s IS ArisCat. Pretože som bol poverený vývojom mobilnej aplikácie pre tento systém, musel som sa so systémom naučiť pracovať a preštudovať funkčnú špecifikáciu systému.

#### 3.2 Štúdie : Možnosti vývoja mobilných aplikácií

Po zoznámení so systémom a jeho funkciami bolo mojou úlohou detailne preštudovať možnosti vývoja mobilných aplikácií, zhodnotiť výhody a nevýhody každej alternatívy a navrhnúť optimálne riešenie.

#### 3.3 Tvorba GUI pre mobilnú aplikáciu v HTML5 Builderi

V tejto časti obdobnej praxe bolo mojou úlohou navrhnúť vhodné riešenie rozloženia prvkov webového IS pre mobilné zariadenia. Práca zahŕňala analýzu a aktívnu konverzáciu s členmi vývojárskeho tímu o výbere tej najvhodnejšej alternatívy. Následne som pokračoval samotným návrhom GUI v HTML5 Builderi.

#### 3.4 Pridávanie funkcionality do vytvoreného GUI

Po vytvorení GUI pre mobilnú aplikáciu bolo mojou úlohou doplniť aplikáciu o rôzne funkčné požiadavky ako napr. povoľovanie/zakazovanie komponentov GUI, zobrazovanie/skrytie komponentov GUI, alebo prechody medzi jednotlivými stránkami aplikácie.

#### 3.5 Testovanie vytvorenej aplikácie na reálnych zariadeniach

V tejto časti odbornej praxe nasledovalo testovanie vytvorenej aplikácie na reálnych zariadeniach. Pre testovanie som mal k dispozícii 2 zariadenia. Vývoj pri testovaní však narazil na veľké množstvo problémov, preto bolo rozhodnuté o zmene IDE, aj o zmene programovacieho jazyka.

#### 3.6 Štúdie: Vývoj pre Android vs. iOS

Na základe neuspokojivých výsledkov testovania na reálnych zariadeniach bolo rozhodnuté o rozdelení vývoja pre jednotlivé mobilné platformy. Mojou úlohou boli detailné

štúdia vývoja mobilných aplikácií pre Google Android a iOS vyhodnotiť výsledky, a zvoliť vhodnú platformu pre vývoj.

### **3.7 Vývoj aktivity pre prihlasovanie**

Mojou prvou úlohou po prechode na programovací jazyk Java a Eclipse IDE bolo vytvoriť aktivitu pre prihlasovanie klientov. Táto aktivita mala takisto na úlohu výber jazyka aplikácie.

### **3.8 Vývoj testovacej aplikácie pre testovanie ovládateľnosti aplikácie**

V tejto časti som pracoval na testovacej aplikácii, ktorá mala za úlohu otestovať ovládateľnosť aplikácie, zobrazenie na rôznych displayoch a zasielanie dát medzi prvkami aplikácie. Aplikácia pracovala so statickými dátami, ale kompletne simulovala zobrazenie a správanie budúcej aplikácie.

### **3.9 Dynamická XML maska**

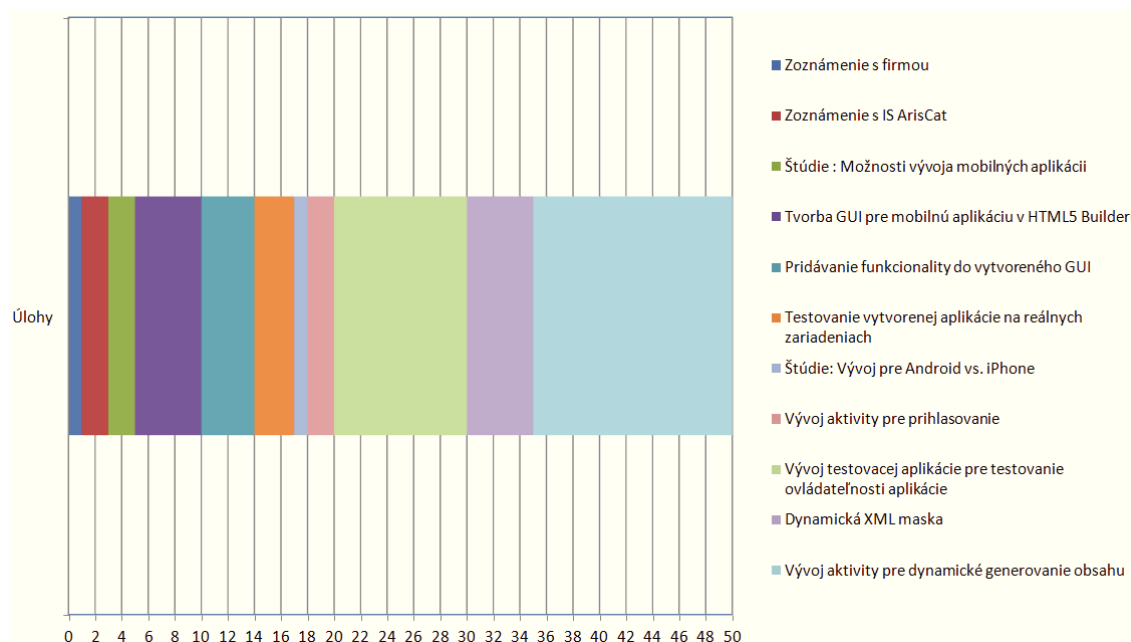
Po úspešnom otestovaní statickej aplikácie vývoj pokračoval definíciou dynamickej XML masky, ktorá bude popisovať jednotlivé komponenty GUI, a na základe tejto masky budú komponenty vytvárané dynamicky počas behu programu. Práca zahŕňala aktívnu konverzáciu s vývojovým tímom o vhodnosti štruktúry XML masky a elementov. Táto časť vývojového procesu zahŕňala takisto štúdium vopred vytvorených XML masiek dodávané firmou.

### **3.10 Vývoj aktivity pre dynamické generovanie obsahu**

Po špecifikácii štruktúry XML masky a jej elementov som mal za úlohu vytvoriť aktivitu, ktorá na základe získanej XML masky z webového serveru vygeneruje príslušné komponenty a doplní ich o funkčnosť. Funkčné požiadavky na komponenty sú takisto definované v získanej XML maske.



### 3.11 Graf časovej náročnosti jednotlivých úloh



Obr. 2: Graf časovej náročnosti jednotlivých úloh



## 4 Používané technológie

Snahou firmy bolo používať produkt firmy Embarcadero, pretože produkty tejto spoločnosti firma už dlhšie používala. Neskôr vývoj pokračoval v open source prostredí Eclipse IDE s Android SDK.

### 4.1 HTML5 Builder

HTML5 Builder je úplným integrovaným vývojovým prostredím pre tvorbu webových a mobilných aplikácií. Využíva technológie HTML5, CSS3 a JavaScript, pretože tieto technológie sú podporované všetkými operačnými systémami a mobilnými platformami. Pre prácu s dynamickými dátami obsahuje HTML5 Builder integrovanú podporu serverových metód v jazyku PHP a podporu všetkých hlavných databázových serverov.

HTML5 Builder umožňuje vytvárať webové aplikácie v celej šírke, teda od klienta až po server, alebo celkom nezávislé mobilné aplikácie pomocou štandardných webových technológií, a nasadiť ich do prevádzky na rôznych operačných systémoch (iOS, Android, Blackberry, Windows Phone). [2]

### 4.2 PhoneGap

PhoneGap je mobilný vývojový framework vyvinutý spoločnosťou Nitobi, a neskôr odkúpený spoločnosťou Adobe Systems. Umožňuje programátorom vytvárať aplikácie pre mobilné zariadenia pomocou jazykov JavaScript, HTML5 a CSS3, namiesto špecifických jazykov jednotlivých mobilných platforiem. Výsledné aplikácie sú hybridné, čo znamená, že nie sú vo svojej podstate natívne (pretože celé rozloženie a vykresľovanie sa vykonáva cez komponent Web View, namiesto natívnych komponentov v rámci UI). Od verzie 1.9 je dokonca možné voľne kombinovať natívny a hybridný kód. [3]

### 4.3 Eclipse IDE

Jedná sa o open source vývojové prostredie určené pre programovanie v jazyku Java. Flexibilita tejto platformy dovoľuje rozšíriť zoznam podporovaných programovacích jazykov pomocou pluginov, napríklad o C++. Zásadný rozdiel medzi Eclipse IDE a ostatnými vývojovými prostrediami v Jave, ako napríklad Netbeans, je práve filozofia úzko zviazaná s rozšíriteľnosťou pomocou pluginov.

V základnej verzii Eclipse neobsahuje napríklad nástroj pre vizuálny návrh grafických užívateľských rozhraní desktopových aplikácií, alebo aplikačný server. Preto vznikli takzvané "subprojekty", ktoré zastrešujú rozšírenia pre jednotlivé oblasti softwarového vývoja v Jave. [4]

## 4.4 Java

Java je objektovo orientovaný programovací jazyk vyvinutý firmou Sun Microsystems. Zdrojový kód jazyka Java je prenositeľný, čo znamená, že kód, ktorý beží na jednej platforme nemusí byť znovu kompilovaný pri spustení na inej platforme. Java aplikácie sú zvyčajne kompilované do tzv. "medzi kódu"(bytecode). Tento formát kódu je nezávislý na architektúre počítača, alebo zariadenia. Program potom pracuje na akomkoľvek počítači, alebo zariadení, ktorý má k dispozícii interpret Javy, tzv. "virtuálny stroj Javy"(Java Virtual Machine).

Syntax jazyka vo veľkej miere pochádza zo syntaxe jazykov C a C++, ale nemá tak nízkoúrovňové vlastnosti ako tieto jazyky. O správu pamäte sa stará automatický Garbage collector, ktorý automaticky vyhľadáva už nepoužívané časti pamäte a uvoľňuje ich pre ďalšie použitie. To je príčinou pomalšieho behu programu v porovnaní s napríklad jazykmi C a C++. [5]

## 4.5 XML

Jedná sa o značkovací jazyk, ktorý definuje súbor pravidiel pre kódovanie dokumentov vo formáte, ktorý je čitateľný pre človeka, aj pre počítač. Jazyk XML nemá preddefinované značky a konkrétne značky sú plne v kompetencii programátora. Hlavnou výhodou je jednoduché definovanie konkrétnych značkovacích jazykov pre rôzne účely a rôzne typy dát. Jazyk je používaný predovšetkým pre výmenu dát medzi aplikáciami a pre publikovanie dokumentov. XML je zjednodušenou podobou staršieho jazyka SGML. Spracovanie XML je podporované veľkým množstvom nástrojov a programovacích jazykov. [6]

## 4.6 Android SDK

Android SDK sú nástroje pre vývojárov softvéru a aplikácií pre platformu Google Android. Obsahuje nástroj pre ladenie programu (debugger), knižnice, emulátor Android zariadenia, dokumentáciu, ukážky kódu a tutoriály. Oficiálnym podporovaným vývojárskym prostredím je Eclipse s použitím pluginu Android Development Tools(ADT). [7]

Na úvod v jednoduchosti vysvetlím niektoré pojmy, ktoré budú spomínané v ďalších častiach bakalárskej práce.

**Activity** - Sú ekvivalentom formulárov, alebo dialógových okien s ovládacími prvkami známymi z iných platforiem. Aplikácia pre Android môže definovať celú radu aktivít. Každá aktivita v rámci Android aplikácie má za úlohu uloženie svojho stavu pri svojom ukončení tak, aby sa k nej v rámci životného cyklu aplikácie dalo opäť vrátiť. Aktivita môže obsahovať prepojenia na iné aktivity. [7]

**TextView** - Zobrazí text užívateľovi a prípadne mu ho umožňuje upraviť. Je v podstate to ekvivalent k Labelu známemu z iných platforiem. [7]

**Spinner** - Poskytuje rýchly spôsob, ako vybrať jednu hodnotu z danej množiny. V predvolenom stave spinner ukazuje na jeho aktuálne zvolenú hodnotu. Po dotyku spinner zobrazí rozbaľovaciu ponuku so všetkými ostatnými dostupnými hodnotami, z ktorých si užívateľ môže vybrať. Je v podstate ekvivalent pre ComboBox známy z iných platforiem. [7]

**Layout** - Každá aktivita má svoj layout, teda základnú definíciu rozvrhnutia grafických komponentov pre danú aktivitu. Napísaný je v XML a je požadované, aby obsahoval aspoň jeden layout. Druhy layoutov je viacero napríklad RelativeLayout, AbsoluteLayout alebo TableLayout . [7]

**Manifest** - Dokument prítomný v každej Android aplikácii. Definuje názov aplikácie, verziu, použité knižnice, použité aktivity, použitú verziu SDK, užívateľské oprávnenia a použité hardwarové prvky. [7]



## 5 Zvolený postup riešenia zadaných úloh

### 5.1 Zoznámenie s IS ArisCAT

Mojou prvou úlohou v rámci odbornej praxe bolo zoznámenie s IS ArisCAT. Vedúci vývoju softwaru mi systém predstavil a priblížil mi jeho funkcionality. Bolo veľmi dôležité poznať systém dokonale, pretože som bol poverený vývojom jeho mobilného ekvivalentu.

*IS ArisCAT alebo Adaptabilný a Robustný Informačný Systém firmy Cathedral Software je kompletný informačný systém pre obchodné a výrobné spoločnosti. Obsahuje rozšírenú možnosť evidencie zákazníkov (CRM), jednoduchú a rýchlu tvorbu ponúk, správu objednávok a riadenie zákazkovej výroby i výroby na sklad. Umožňuje viesť neobmedzené množstvo skladov a organizovať materiál a tovar do skladových skupín. Manažérske výstupy a štatistické dáta je možné ľubovoľne upravovať a exportovať do dokumentov Microsoft Office. Užívateľ programu si môže sám meniť tlačové zostavy a masky pre konfiguráciu výrobkov.*  
(Cathedral Software, 2013) [8]

IS ArisCAT je v podstate dynamický informačný systém pre predajcov a obchodníkov, kde obsah systému závisí na danom predajcovi. Systém umožňuje vytvárať objednávky, zobrazuje zákazníkovi skalkulované ponuky, postup realizácie zákaziek, expedíciu, vystavené faktúry a ich obrat a platobnú morálku.

Obr. 3: Zadávanie objednávky vo webovom IS ArisCAT

## 5.2 Štúdie : Možnosti vývoja mobilných aplikácií

Mojou úlohou bolo detailne preštudovať možnosti vývoja mobilných aplikácií, zhodnotiť výhody a nevýhody každej alternatívy, a navrhnúť optimálne riešenie. Snahou firmy bolo zasiahnuť všetky možné mobilné platformy. Túto funkcionality ponúkal Embarcadero HTML5 Builder, na ktorý mala firma zakúpenú licenciu. Preto som sa zameral najmä na porovnanie vývoja v už spomínanom HTML5 Builderi a vývoja natívnych Android aplikácií. Vyhodnotil som detailnú správu, a po konverzácii a zhodnotení požiadaviek firmy bolo rozhodnuté, že vývojová platforma bude práve HTML5 Builder.

## 5.3 Tvorba GUI pre mobilnú aplikáciu v HTML5 Builderi

V tejto časti obdobnej praxe bolo mojou úlohou navrhnúť vhodné riešenie rozloženia prvkov webového IS pre mobilné zariadenia. Po mojom návrhu pokračovala konverzácia s členmi vývojárskeho tímu o vhodnosti, respektíve nevhodnosti rozloženia prvkov. Hlavným problémom bolo, ako vhodne a čitateľne rozložiť prvky webového IS pre omnoho menšie displaye mobilných zariadení tak, aby bolo GUI user-friendly. Užívateľské rozhranie sme niekoľkokrát analyzovali a navrhovali na papier vo forme náčrtu, až sme dospeli k vhodnej alternatíve. Po schválení návrhu členmi vývojárskeho tímu som začal pracovať na samotnom návrhu GUI v HTML5 Builderi. Išlo o vytvorenie GUI podľa náčrtov, ktoré zahŕňali rôzne časti webového IS, ako napríklad prihlasovaciu obrazovku, menu, zoznam zákaziek, či novú objednávku. Aplikáciu som naplnil statickými dátami dostupnými z webového IS tak, aby vzhľad GUI zodpovedal požadovanému vzhľadu budúcej aplikácie.

## 5.4 Pridávanie funkcionality do vytvoreného GUI

Po úspešnom vytvorení GUI pre mobilnú aplikáciu bolo mojou úlohou doplniť aplikáciu o rôzne funkčné požiadavky. Tie boli tri a zahŕňali prechody medzi stránkami GUI, povolenie/zakazovanie prvkov a zobrazenie/skrytie prvkov. Ako prvé som sa rozhodol riešiť prechody. Bolo to jednoduché presmerovanie aplikácie na ďalšiu stránku tak, ako bolo špecifikované vo webovom IS. Po splnení tejto požiadavky už bolo možné si aplikáciu otestovať vo webovom prehliadači. Spolu s vedúcim vývoju softwaru sme aplikáciu otestovali, okomentovali, a pokračoval som na splnení ďalších požiadaviek. Išlo o nasimulovanie správania webového IS, tj. pri kliku na určitú položku ComboBoxu sa vykonala určitá akcia, ktorá manipulovala s prvkami GUI.



---

```
<script type="text/ javascript ">

    function makeUnvisible( elementID )
    {
        document.getElementById( elementID ).style.visibility = "hidden";
    }

    function makeVisible( elementID )
    {
        document.getElementById( elementID ).style.visibility = " visible ";
    }

</script>
```

---

Výpis 1: Ukážka funkcií využívaných pre zobrazovanie/skrytie elementov GUI

---

```
<script type="text/ javascript ">

    function makeDisabled( elementID )
    {
        document.getElementById( elementID ).disabled=true;
    }

    function makeEnabled( elementID )
    {
        document.getElementById( elementID ).disabled=false;
    }

</script>
```

---

Výpis 2: Ukážka funkcií využívaných pre povoľovanie/zakazovanie elementov GUI

---

## 5.5 Testovanie vytvorenej aplikácie na reálnych zariadeniach

Po úspešnom otestovaní statickej aplikácie vo webovom prehliadači pokračovalo testovanie na reálnych zariadeniach. Pri testovaní som mal k dispozícii 2 zariadenia.

Prvým bolo HTC Desire s operačným systémom Google Android verzie 2.3.3, a druhým bol Apple iPhone 4 s operačným systémom iOS verzie 4.0.2.

Pri snahe testovať aplikáciu na týchto zariadeniach som však narazil na veľké množstvo problémov. ComboBoxy zobrazovali nekorektné informácie a nezobrazovali miniatúry obrázkov, ktoré však boli v požiadavkách na aplikáciu. Pri nahraní aplikácie do mobilu, ešte pred jej spustením, zobrazovalo nepochopiteľné upozornenia. HTML5 Builder prepisoval údaje, ktoré boli zadané (napríklad popis elementu - Caption ). V zariadení pri snahe prejsť na ďalšiu stránku aplikácia spadla a podobne. Po konverzácii s vedúcim vývoju softwaru a podrobnej analýze sme na riešenie neprišli, a preto sa rozhodlo, že vývoj v HTML5 Buildri sa ukončí a rozdelí sa medzi jednotlivé mobilné platformy. Práve

v tejto časti praxe som najviac pocítil to, že vývoj softwaru nie vždy napreduje podľa predstáv a bez problémov.

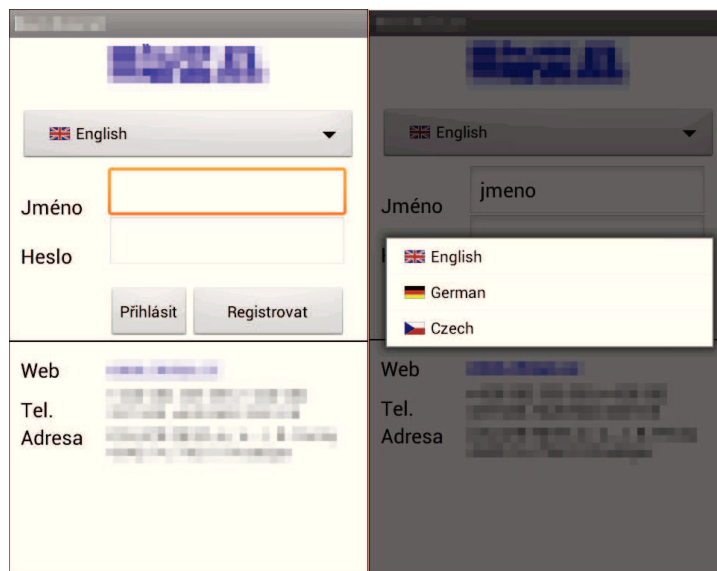
## 5.6 Štúdie: Vývoj pre Android vs. iOS

Po rozhodnutí o rozdelení vývoja pre jednotlivé mobilné platformy bolo mojou úlohou preštudovať možnosti vývoja pre Google Android a iOS. Zozbieral som potrebné dáta a vytvoril dokument popisujúci vývoj pre tieto platformy. Po konzultácii s vedúcim vývoju softwaru bolo rozhodnuté pre vývoj pre Google Android. Rozhodujúcim faktorom bolo zastúpenie na trhu (Google Android - 68,3 %, iOS 18,8 %. Údaje pre rok 2012) ako aj to, že pri vývoji pre iOS by bol potrebný nový hardware. Nevýhodou tohto rozhodnutia bolo to, že systém nebude dostupný pre užívateľov iOS.

## 5.7 Vývoj aktivity pre prihlasovanie

Po prechode na platformu Google Android bolou mojou prvou úlohou vytvoriť prihlasovaciu aktivitu aplikácie. Aktivita mala obsahovať textové polia pre meno a heslo, spinner pre výber jazyka aplikácie, tlačidlá pre prihlásenie a registráciu, a informácie o firme.

Ako prvé bolo potrebné zvoliť typ layoutu. Zvolil som RelativeLayout z dôvodu nerovnomerného rozloženia komponentov. Prvé textové pole slúži pre zadanie mena užívateľa, druhé pre jeho heslo, ktoré je vyplňované bodkami vďaka nastaveniu vlastnosti `android:password="true"` pri prvku `EditText` v XML súbore. Ošetrenie hesla má na starosti podmienka `if-else`, v ktorej porovnávam obsah polí s pevne nastaveným menom a heslom. Pre informácie o firme som zvolil komponent `TextView`.



Obr. 4: Prihlasovacia aktivita

Spinner pre výber jazyka musel zobrazovať miniatúru vlajky krajiny a jej názov, preto som zvolil vlastný ArrayAdapter. ArrayAdapter je objekt, ktorý definuje štýl spinneru a pole údajov, ktoré budú na výber. Keďže mnou vytvorená trieda CountryAdapter musela dediť z ArrayAdapter<String>, tak následne musela vo svojom tele aj prepisovať metódy tejto triedy. Sú to metódy getDropDownView a getView. Metóda getView popisuje zobrazenie spinneru pred rozbalením a metóda getDropDownView popisuje zobrazenie spinneru po rozbalení. Obe metódy vracajú objekt typu View, teda presne to, čo užívateľ uvidí na displayi zariadenia. Priradenie vytvoreného adaptéru príslušnému spinneru zabezpečuje metóda setAdapter triedy Spinner, v mojom prípade spCountry.setAdapter(new CountryAdapter());.

## 5.8 Vývoj testovacej aplikácie pre testovanie ovládateľnosti aplikácie

Po dokončení prihlasovacej aktivity som pokračoval vo vývoji ďalších vybraných častí systému. Ako prvé som navrhoval hlavné menu aplikácie, ktoré bolo tvorené objektmi typu ImageView. Objekt ImageView rieši zobrazenie obrázkov v layoute.



Obr. 5: Hlavné menu aplikácie

Po vytvorení hlavného menu som pracoval na vývoji aktivít pre jednotlivé výrobky predajcu. Išlo o viacero aktivít, z ktorých každá reprezentovala iný výrobok predajcu. Pre tieto aktivity som zvolil TableLayout, pretože prvky sú zoradené pod sebou a pre formulárové aplikácie je tento layout najvhodnejší. TableLayout obsahoval množstvo objektov typu TableRow, ktoré reprezentujú riadky tabuľky. Každý TableRow obsahoval v ľavej bunke TextView s popisom komponentu a v pravej samotný komponent GUI typu Spinner, alebo EditText. Po návrhu GUI aktivít výrobkov a naplnení spinnerov dátami som

pokračoval v ďalšej funkcionalite, konkrétne povoľovanie a zakazovanie komponentov GUI. Riešenie tejto funkcionality ponúkol listener triedy AdapterView, a to konkrétne OnItemSelectedListener. Tento listener má za úlohu vykonať akciu pri výbere položky zo spinneru. Priradenie listenera k danému spinneru zabezpečuje metóda setSelectedListener triedy Spinner.

Po skompletizovaní tejto funkčnosti nasledovalo testovanie ovládateľnosti aplikácie na reálnom zariadení. K dispozícii som mal 2 zariadenia.

Prvým bolo HTC Desire s operačným systémom Google Android verzie 2.3.3, a druhým HTC One X s operačným systémom Google Android verzie 4.0..

Pri testovaní mi vedúci vývoju softwaru, ako aj ostatní členovia programátorského tímu, dávali pripomienky, ktoré som si zaznamenal a upravil aplikáciu podľa nich.

## 5.9 Dynamická XML maska

V tejto časti odbornej praxe bolo mojou úlohou nastudovať si štruktúru XML masky výrobku. Každý výrobok má jedinečnú XML masku skladajúcu sa z objektov rôznych typov. Po nastudovaní nasledovala konverzácia s vedúcim vývoju softwaru a členom programátorského tímu, ktorý má v kompetencii práve túto problematiku. Išlo o hľadanie riešenia ako upraviť XML masku z webového IS tak, aby bola vhodná aj pre mobilné zariadenia. Preto sa museli objekty v XML dokumente zoradiť do takého poradia, ako budú zobrazené na mobilnom zariadení, a celkom sa vypustili objekty typu Label.

Popisy k jednotlivým komponentom GUI zabezpečoval atribút DisplayName tagu Object.

**Masks** - koreňový tag = iba jeden výskyt (XML obal)

**Mask** - Rodič, definícia jednej dynamickej masky. Obecne sa môže tento tag v XML opakovať viackrát v prípade, že predané XML popisuje viac dynamických masiek na žiadosť volajúceho (prakticky však XML takmer vždy popisuje iba jednu masku). **Object** - Je jeden grafický element (v Java kóde ho zastupuje objekt TableRow). Podľa hodnoty atribútu TypLECG môže nadobúdať rôzne typy komponentov GUI.

**Item** - Položka grafického elementu(Týka sa len objektov typu 'C' - ComboBox)

**Action** - Je to akcia (trigger), ktorá sa má spustiť pri výbere danej položky (Item), ktorej je tento tag dieťaťom. Jeden záznam akcie môže ovplyvniť iba jeden cieľový objekt (Object). Jedna položka (Item) však môže mať viac takýchto akcií (Action) preto, aby výbraním istej položky (Item) došlo k ovplyvneniu viacerých objektov (Object).

Ukážka XML masky je umiestnená v prílohe A.

## 5.10 Vývoj aktivity pre dynamické generovanie obsahu

Poslednou úlohou v rámci mojej odbornej praxe bol vývoj aktivity, ktorej obsah a komponenty sú vygenerované za behu programu, na základe dodanej XML masky. Táto úloha dopĺňala aj prihlasovaciu aktivitu. To znamená, že na základe mena a hesla užívateľa sa vygeneruje príslušný obsah aplikácie. V prvej časti, po kliknutí na tlačidlo Prihlásiť, aplikácia pošle HTTP post na webový server spolu s informáciou o užívateľových údajoch. PHP skript, uložený na webovom serveri prijme informáciu o užívateľových údajoch, spracuje ju, a na základe podmienky pošle príslušnú XML masku vo forme textu. Text je následne zaslaný pomocou triedy Intent do ďalšej aktivity.

Android aplikácia nemá štandardne povolenú sieťovú komunikáciu. Je ju potrebné povoliť, a to pridaním povolenia do Android Manifestu.

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Po získaní textovej reprezentácie XML masky v nasledujúcej (dynamicky generovanej) aktivite bolo potrebné z nej vytvoriť XML dokument, aby sa s ním dalo ďalej pracovať.

---

```
public Document XMLfromString(String xml) {
    Document doc = null;
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    try {
        DocumentBuilder db = dbf.newDocumentBuilder();
        InputSource is = new InputSource();
        is.setCharacterStream(new StringReader(xml));
        doc = db.parse(is);
    } catch (ParserConfigurationException e) {
        System.out.println("XML_parse_error:_" + e.getMessage());
        return null;
    } catch (SAXException e) {
        System.out.println("Wrong_XML_file_structure:_" + e.getMessage());
        return null;
    } catch (IOException e) {
        System.out.println("I/O_exception:_" + e.getMessage());
        return null;
    }
    return doc;
}
```

---

Výpis 3: Ukážka metódy generujúcej XML dokument

Po získaní XML dokumentu bolo potrebné dané dáta z dokumentu získať, a na základe ich hodnoty vygenerovať komponenty GUI. O to sa starala metóda, ktorá prešla všetky tagy typu Objekt a na základe hodnoty riadiaceho atribútu vygenerovala prí-

slušné komponenty alebo skupiny komponent do masky.

Po vygenerovaní GUI bolo potrebné nastaviť akcie pre výber niektorej položky zo spinneru. To opäť zaistí ovládanie `OnItemSelectedListener`. Ten zistí, o ktorý spinner z GUI ide, na ktorú položku bolo kliknuté a získal tagy typu `Action` danej položky. Tagy boli následne spracované a bola vykonaná definovaná akcia.

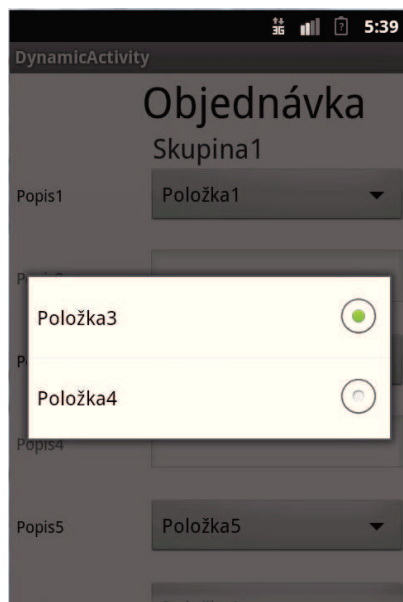
Jednou z požiadaviek bolo aj zobrazovanie číselnej softwarovej klávesnice namiesto klasickej, pri `EditText`och s atribútom `EditTextType` rovným 'I', alebo 'F'. To znamená, že ak do poľa `EditText` je žiadaná číselná hodnota, zobrazí sa číselná klávesnica, ak je žiadaný text, zobrazí sa klasická QWERTY klávesnica.

```
edText.setInputType(InputType.TYPE_CLASS_NUMBER);
```

Po kompletnom naprogramovaní funkcionality úloha pokračovala testovaním. Tentokrát som mal k dispozícii 3 zariadenia.

Prvým bolo HTC Desire s operačným systémom Google Android verzie 2.3.3, druhým HTC One X s operačným systémom Google Android verzie 4.0., a tretím bol tablet Samsung Galaxy Tab 10.1 s operačným systémom Google Android verzie 3.0..

Pretože veľkosť dynamicky generovaných komponentov bola zadávaná pomerom k veľkosti displaya zariadenia, bolo zobrazenie korektné na oboch telefónoch s rozdielnymi veľkosťami displayov, ako aj na omnoho väčšom tablete.



Obr. 6: Výsledný vzhľad dynamicky vygenerovanej aktivity

## 6 Potrebné znalosti získané počas štúdia potrebné pre výkon odbornej praxe

Pre potreby výkonu praxe som potreboval znalosti z predmetov týkajúcich sa programovania a užívateľských rozhraní.

Na začiatku praxe som pracoval v HTML5 Buildri, ktorý využíval HTML, CSS a JavaScript. V tejto časti mi veľmi pomohol predmet Vývoj internetových aplikáci (**VIA**).

V predmete Vývoj internetových aplikáci, ktorý som navštevoval paralelne s vykonávaním praxe, boli prebraté jazyky HTML, CSS aj JavaScript, ktoré som pri práci využíval. Na predmete som získal mnoho poznatkov, hlavne ohľadom JavaScriptu, bez ktorých by som musel veľkú časť praxe venovať samoštúdiu.

V druhej časti praxe som programoval pre operačný systém Google Android. Pre tento operačný systém sa programuje v jazyku Java, do ktorého som prenikol vďaka predmetu Programovací Jazyky I (**PJ I**). Počas tohto predmetu som si Javu veľmi obľúbil a získal veľké množstvo znalostí, ktoré som pri praxi použil. V Jave som pokračoval aj zvolením predmetov Tvorba aplikácií pro mobilné zařízení II (**TAMZ II**) a Java technologie (**JAT**).

V predmete TAMZ II som prvý získal prvé skúsenosti s programovaním pre Android a postupom semestra, pri riešení zložitejších úloh, som nazbieral množstvo skúseností, bez ktorých by som sa v druhej časti praxe nezaobišiel. Tento predmet bol pre moju prax najdôležitejší.

V predmete JAT som si prehľbil znalosti Javy o ďalšie technológie ako Servlety, JSF, alebo EJB. Tento predmet mi už definitívne potvrdil smer, ktorým chcem ako vývojár pokračovať - a to programovanie v jazyku Java.

Takisto v predmete Vývoj informačných systémov (**VIS**) som projekt programoval v Jave, a opäť prehľbil znalosti v tomto programovacom jazyku.

Z pohľadu designu som využil teoretické znalosti z predmetu Užívateľská rozhraní (**URO**), kde bolo cieľom vytvoriť desktopové aplikácie v jazyku Python a Java s dôrazom na užívateľské rozhranie.





## 7 Odhalené nedostatky počas absolvovania praxe

Ako môj hlavný nedostatok som pociťoval nedostatok skúseností s prácou v reálnej firme. Začal som si uvedomovať, že to, na čom pracujem, nie je školský projekt, ale reálna aplikácia, ktorú budú ľudia používať. Po čase som sa však aklimatizoval a na firemné prostredie som si zvykol. Ďalšie nedostatky som pociťoval v tom, že som pred praxou neovládal jazyk PHP, s ktorým som musel v niektorých fázach praxe pracovať. Jazyk som sa však postupne učil a dané požiadavky plnil. Takisto som nemal skúsenosti s HTML5 Buildrom, tak som si musel na nové prostredie dlhšie zvykať a neskôr pri problémoch som zistil, že projekt nie vždy pokračuje podľa plánov a predstáv. Vo všetkých fázach vývoja však boli zamestnanci nápomocní, a spolu s mojou chuťou učiť sa nové veci som sa cez tieto problémy preniesol.



## 8 Hodnotenie bakalárskej praxe

Počas odbornej praxe som sa naučil množstvo nových poznatkov ako používať vývojové nástroje a pracovať na reálnom softvérovom projekte. Naučil som sa tímovo pracovať a aktívne konzultovať o riešených problémoch, ale aj samostatnej práci v neznámych technológiách. Z daných dôvodov hodnotím odbornú prax pozitívne. Práca s ostatnými zamestnancami bola veľmi motivujúca a získané odborné poznatky zvýšia moje šance pri hľadaní práce v tomto odbore.

Uvedomil som si, že mi škola poskytla kvalitný základ, na ktorom môžem ďalej stavať a rozvíjať svoje profesijné znalosti tak, aby som bol schopný splňať dnešné požiadavky neustále meniaceho sa trhu.

Michal Walder



## 9 Literatúra

- [1] Cathedral Software. [online]. 2013 [cit. 2013-05-06].  
Dostupné z URL: <[www.cathedral.cz](http://www.cathedral.cz)>
- [2] HTML5 Builder. [online]. 2013 [cit. 2013-05-06].  
Dostupné z URL: <<http://www.embarcadero.com/products/HTML5-Builder>>
- [3] PhoneGap. [online]. 2013 [cit. 2013-05-06].  
Dostupné z URL: <<http://phonegap.com/about/>>
- [4] Eclipse IDE. [online]. 2013 [cit. 2013-05-06].  
Dostupné z URL: <<http://www.eclipse.org/org/>>
- [5] POORNACHAMDRA, Sarang, *Java Programming: Learn Advanced Skills from a Java Expert*. 1. 2012. ISBN-10: 978-0-07-163360-1
- [6] RAY, Eric T., *Learning XML, Second Edition*. 2. 2003. ISBN-10: 0-596-00420-6
- [7] MEIER, Reto, *Professional Android 4 Application Development*. 3. 2012. ISBN: 978-1-118-10227-5
- [8] ArisCAT. [online]. 2013 [cit. 2013-05-06].  
Dostupné z URL: <<http://www.cathedral.cz/cz/m/o-systemu/>>



## 10 Prílohy

Mimo príloh, na ktoré je v textovej časti priamy odkaz, je na CD uložený aj dokument s ukážkami niektorých zdrojových kódov pre aplikáciu.

**Príloha A – Ukážka XML dynamickej masky.**

**Príloha B - Dokument s ukážkami vybraných úsekov zdrojových kódov aplikácie.**